The Geometry of LLM Quantization: GPTQ as Babai's Nearest Plane Algorithm

Jiale Chen[IST Austria] Yalda Shabanzadeh[IST Austria] Torsten Hoefler[ETH Zürich] Dan Alistarh[IST Austria]

Summary

Large language models (LLMs) are typically stored in 16-bit precision, which makes deploying them expensive in terms of memory and computation. Post-training weight quantization, especially down to around 4-bit, has become the de facto way to run GPT-scale models on affordable hardware. Among such methods, GPTQ is one of the most widely used one-shot quantizers and remains competitive even with more recent techniques.

Despite its popularity, GPTQ is usually presented as a sequence of algebraic updates with little geometric intuition and no clear worst-case guarantees. In this work, we show that GPTQ has a precise geometric interpretation: when applied to a linear layer from the last dimension to the first, GPTQ is mathematically equivalent to **Babai's nearest plane algorithm** for the **closest vector problem (CVP)** on a lattice induced by the input Hessian.

This equivalence lets us (1) view GPTQ as an orthogonal walk through a sequence of affine subspaces in activation space, and (2) directly import classical error bounds from lattice algorithms. Beyond analysis, we also explore how this viewpoint interacts with quantization order and entropy-constrained (Huffman) coding, enabling highly compressed yet accurate LLMs.

GPTQ, Closest Vector Problem (CVP), and Babai

Algorithm 1: GPTQ

Input: $\boldsymbol{W}, \boldsymbol{S}, \boldsymbol{X}, \boldsymbol{T}, \lambda, \mathbb{Z}^{\dagger}$
Output: $\boldsymbol{Z}, \boldsymbol{Q}$

$\boldsymbol{H} \leftarrow \boldsymbol{T}^{\top}\left(\boldsymbol{X}^{\top}\boldsymbol{X} + \lambda \mathbf{I}\right)\boldsymbol{T}$
$\boldsymbol{L} \leftarrow \mathrm{LDL}\left(\boldsymbol{H}^{-1}\right)$
$\boldsymbol{W}, \boldsymbol{S} \leftarrow \boldsymbol{T}^{-1}\boldsymbol{W}, \boldsymbol{T}^{-1}\boldsymbol{S}$
$\boldsymbol{Q}, \boldsymbol{Z} \leftarrow \boldsymbol{W}, \mathbf{0}$
for $j \leftarrow 1$ to c **do**
  $\zeta \leftarrow \boldsymbol{W}[j,:]/\boldsymbol{S}[j,:]$
  $\boldsymbol{Z}[j,:] \leftarrow \textsc{Round}\left(\zeta, \mathbb{Z}^{\dagger}\right)$
  $\boldsymbol{Q}[j,:] \leftarrow \boldsymbol{Z}[j,:] * \boldsymbol{S}[j,:]$
  $\varepsilon \leftarrow \boldsymbol{Q}[j,:] - \boldsymbol{W}[j,:]$
  $\boldsymbol{W}[j:,:] \leftarrow \boldsymbol{W}[j:,:] + \boldsymbol{L}[j:,j]\varepsilon$
end
$\boldsymbol{Z}, \boldsymbol{Q} \leftarrow \boldsymbol{TZ}, \boldsymbol{TQ}$

Algorithm 2: Babai's Nearest Plane

Input: $\boldsymbol{B}, \boldsymbol{y}$
Output: $\boldsymbol{z}$

$\boldsymbol{T} \leftarrow \mathrm{LLL}\left(\boldsymbol{B}\right)$ `// transformation`
$\boldsymbol{A} \leftarrow \boldsymbol{BT}$ `// basis reduction`
$\Phi \leftarrow \mathrm{QR}\left(\boldsymbol{A}\right)$ `// orthogonalize`
$\boldsymbol{y}', \boldsymbol{z} \leftarrow \boldsymbol{y}, \mathbf{0}$
for $j \leftarrow c$ to 1 **do**
  $\zeta \leftarrow \langle \Phi[:,j], \boldsymbol{y}' \rangle / \langle \Phi[:,j], \boldsymbol{A}[:,j] \rangle$
  $\boldsymbol{z}[j] \leftarrow \textsc{Round}\left(\zeta, \mathbb{Z}\right)$
  $\boldsymbol{y}' \leftarrow \boldsymbol{y}' - \boldsymbol{A}[:,j]\boldsymbol{z}[j]$
end
$\boldsymbol{z} \leftarrow \boldsymbol{Tz}$

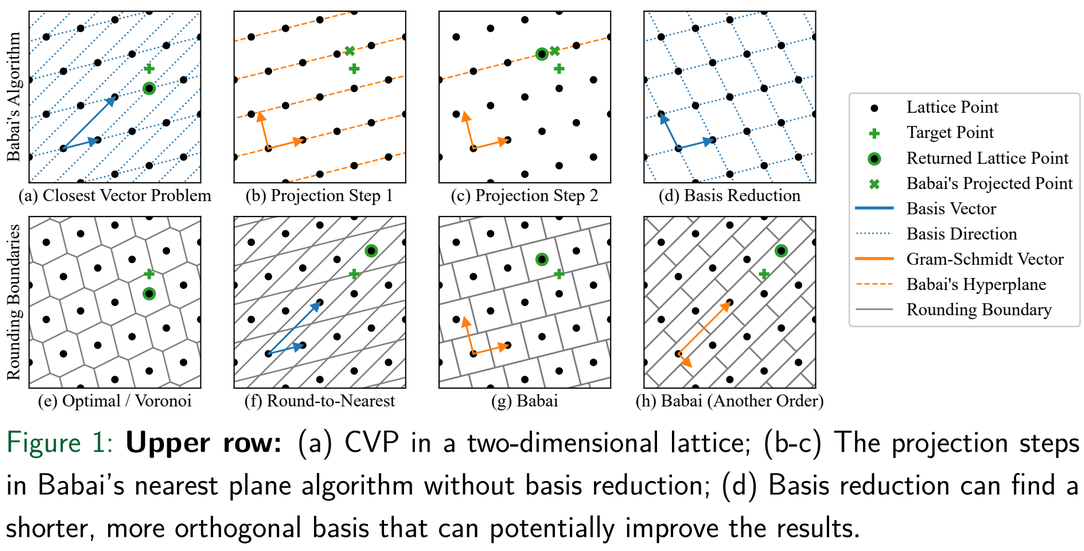

Figure 1: **Upper row:** (a) CVP in a two-dimensional lattice; (b-c) The projection steps in Babai's nearest plane algorithm without basis reduction; (d) Basis reduction can find a shorter, more orthogonal basis that can potentially improve the results.
Lower row: rounding boundaries of (e) optimal rounding or Voronoi cells; (f) round-to-nearest (RTN); (g) Babai's nearest plane algorithm without basis reduction; (h) Babai's algorithm without basis reduction under reversed basis ordering.

Equivalence: GPTQ/OBQ $\Leftrightarrow$ Babai

GPTQ and Babai's algorithm have the same results if we align the dimensional order of these two algorithms, e.g., running GPTQ from the last to the first dimension.

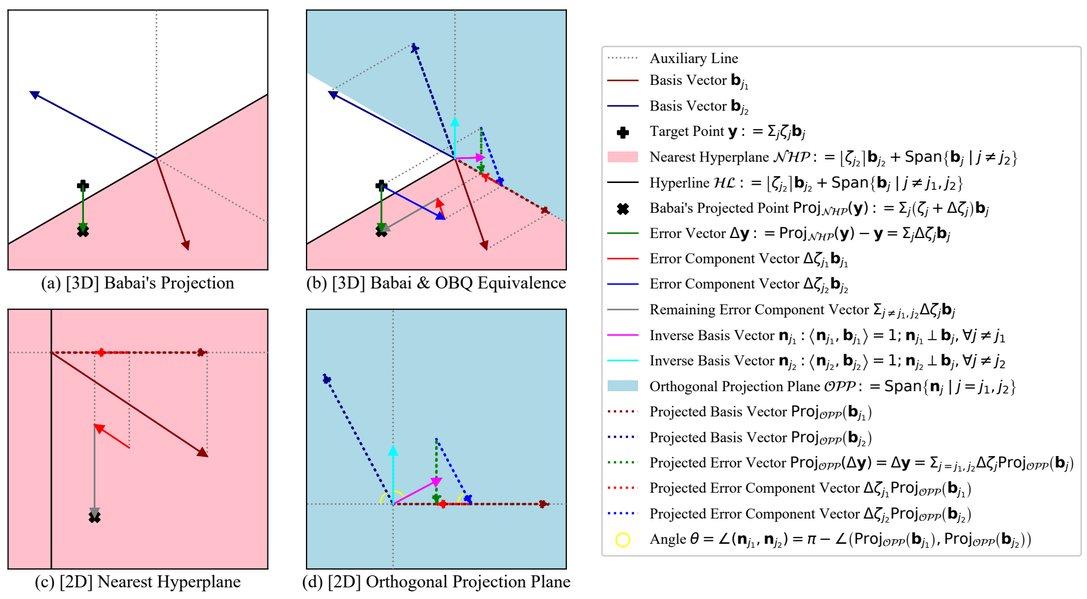

Figure 2: Equivalence of OBQ's error propagation and Babai's projection. (a) 3D plot showing the target point being projected onto the nearest plane. (b) 3D plot showing how the projection error is propagated. (c) 2D plot showing the vectors on the nearest hyperplane in (b). (d) 2D plot showing the vectors on the orthogonal projection plane in (b).

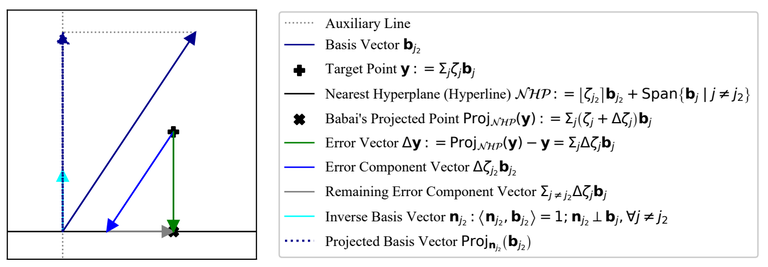

Figure 3: Geometric interpretation of OBQ's quantization order. This 2D plot shows the target point being projected onto the nearest plane.

GPTQ Error Bound

Assume there is no clipping ($\mathbb{Z}^{\dagger} = \mathbb{Z}$). Let $\boldsymbol{D}$ be the diagonal matrix in the LDL decomposition of the Hessian matrix $\boldsymbol{X}^{\top}\boldsymbol{X}$. For every output channel i ($1 \leq i \leq r$) produced by Babai's algorithm, or equivalently GPTQ executed back-to-front, the quantization error has a tight error upper bound: $\left\|\boldsymbol{X}\,\mathrm{diag}\left(\boldsymbol{s}_i\right)\boldsymbol{z}_i - \boldsymbol{X}\boldsymbol{w}_i\right\|^2 \leq \frac{1}{4}\boldsymbol{s}_i^{\top}\boldsymbol{D}\boldsymbol{s}_i$.

Clip-Free GPTQ via Huffman Encoding

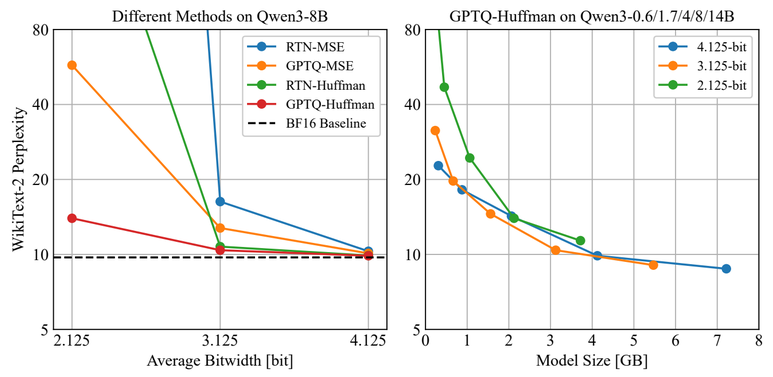

Figure 4: Perplexity compression trade-offs for Qwen3 models under different quantization schemes. **Left:** Comparison of quantization methods (RTN-MSE, GPTQ-MSE, RTN-Huffman, and GPTQ-Huffman) on Qwen3-8B evaluated on WikiText-2. Perplexity is plotted against the average effective bitwidth per weight, with the BF16 baseline shown as a dashed line. GPTQ-Huffman has the best (lowest) perplexity. **Right:** Scaling behavior of GPTQ-Huffman across multiple model sizes (0.6B, 1.7B, 4B, 8B, 14B) and bitwidths (4.125, 3.125, 2.125). The x-axis denotes the effective model size after quantization, and the y-axis shows perplexity on WikiText-2. Each curve corresponds to a fixed bitwidth, while points along a curve represent different model scales. Using our GPTQ-Huffman method, 3.125-bit stands out as the Pareto optimal bitwidth.

NeurIPS 2025 Workshop on Symmetry and Geometry in Neural Representations (NeurReps 2025)